

## ЛАБОРАТОРНАЯ РАБОТА № 7

### Классы и методы

#### 1. Краткие теоретические сведения

Класс является обобщенным понятием, определяющим характеристики и поведение некоторого множества конкретных объектов этого класса, называемых экземплярами, или объектами, класса.

В целях ускорения работы программиста и упрощения применяемых алгоритмов применяют классы. Применение классов позволяет разбить сложную задачу на ряд простых, взаимосвязанных задач. Прodelав эту операцию неоднократно применительно к вновь полученным классам, можно получить алгоритм решения задачи в виде набора простых, понятных, легко осуществимых классов, взаимодействующих друг с другом.

«Классический» класс содержит данные, задающие свойства объектов класса, и функции, определяющие их поведение. В последнее время в класс часто добавляется третья составляющая – события, на которые может реагировать объект класса. Это оправдано для классов, использующихся в программах, построенных на основе событийно-управляемой модели, например, при программировании для Windows.

Процедура объявления и создания экземпляров пользовательского типа не отличается от таковой для предопределенных типов за исключением необходимости использования Спецификатора New (new) для создания экземпляров пользовательских типов значений и ссылочных типов.

## Спецификаторы классов:

1. **new** – используется для вложенных классов. Задаёт новое описание класса взамен унаследованного от предка. Применяется в иерархиях объектов.

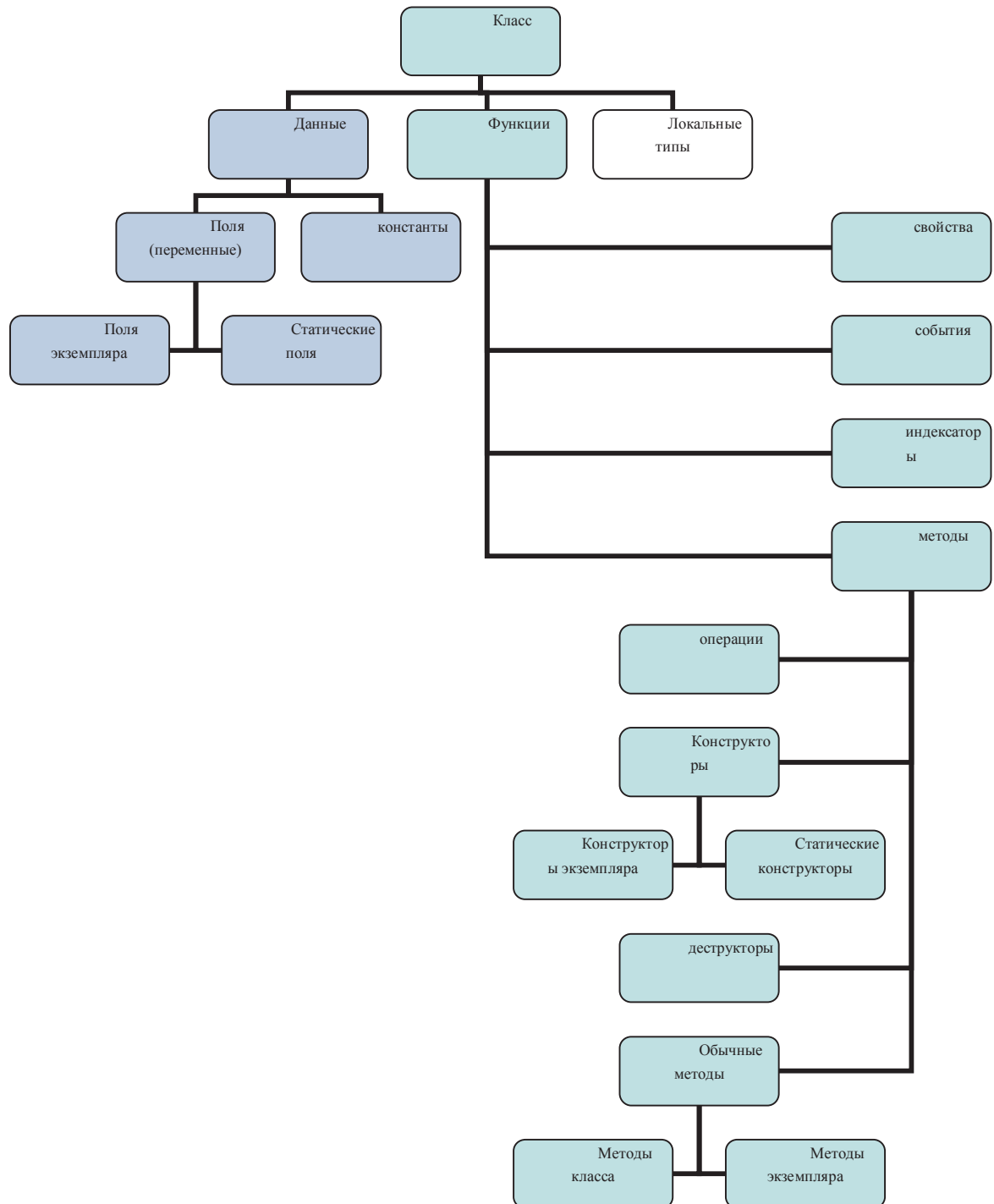


Рис. 7.1. Класс

2. **public** – доступ не ограничен
3. **protected** – используется для вложенных классов. Доступ только из элементов данного и производных классов
4. **internal** – доступ только из данной программы (сборки)
5. **protected internal** – доступ только из данного и производных классов или из данной программы (сборки)
6. **private** – используется для вложенных классов. Доступ только из элементов класса, внутри которого описан данный класс
7. **abstract** – абстрактный класс. Применяется в иерархиях объектов.
8. **sealed** – бесплодный класс. Применяется в иерархиях объектов.
9. **static** – статический класс. Введен в версию языка 2.0.

Спецификаторы 2 – 6 называются спецификаторами доступа. Они определяют, откуда можно непосредственно обращаться к данному классу. Спецификаторы доступа могут присутствовать в описании только в вариантах, приведенных в таблице, а также могут комбинироваться с остальными спецификаторами.

## Методы

Методы делают всю работу, для исполнения которой предназначены классы и структуры: вычисляют значения, изменяют данные, получают вводимую информацию – в общем, выполняют все действия, составляющие поведение объекта

## Объявление методов

Один из видов членов, которые добавляются к классам, – это методы, определяющие действия, которые должен выполнять

класс. Есть две разновидности методов: те, что возвращают значения, и те, которые никаких значений не возвращают. Следующий пример иллюстрирует обе разновидности методов:

```
// Этот метод возвращает целые (int) значения
```

```
public int Add (int first, int second)
```

```
{
```

```
    int Result;
```

```
    Result = first + second;
```

```
    Return result;
```

```
}
```

```
public void myVoidMethod ()
```

```
// void – не возвращает значений
```

```
{
```

```
    MessageBox.Show ("Этот метод не возвращает  
    значения ");
```

```
}
```

Метод является самостоятельной частью кода, которая имеет имя и может содержать аргументы, выполнять последовательность инструкций (операторов) и изменять значения своих аргументов.

### **Вызов методов.**

Метод не выполняется, пока его не вызовут. Чтобы вызвать метод, следует указать его имя и задать необходимые ему параметры.

```
// Так вызывают метод Rotate с двумя параметрами
```

```
Rotate (45, "Degrees");
```

Метод Main – особый, он вызывается сразу после начала исполнения программы. К особым методам также относятся деструк-

торы, которые вызывают во время выполнения непосредственно перед уничтожением объекта. Третий вид особых методов – конструкторы; они исполняются во время инициализации объектов.

Метод может иметь один или несколько параметров. Параметр – это аргумент, передаваемый методу вызывающим его методом. При объявлении метода список параметров указывают после его имени в круглых скобках, при этом для каждого параметра необходимо указать его тип.

```
public void DisplayName (string name, byte age)
{
    Console.WriteLine ("Hello " + name + ". You are "
        + age.ToString () +"years old.");
}
```

**Внимание.** У этого метода два параметра: один типа `String`, с локальным именем `name`, а второй, `age`, – типа `Byte`. Их область видимости ограничена методом, вне которого переменные не доступны.

Составим фрагмент программы вычисления следующих функций:

$$C = \frac{4,5e^{x^2}}{e^x + 8,5 \sin(x)} \quad \text{и} \quad D = \frac{1}{8e^{x^4} - \sin(x^4) + x^3}$$

Метод `fnc` типа `double`

```
public double fnc(a as double, x as double, z double)
{
    return (a*exp(x) + z * sin(x));
}
```

Использование метода

```

. . .
c = fnc ( a, x*x, 0. ) / fnc ( 1., x, 8.5 )
b = -1.
d = 1/( fnc (8., x^4, b) + x ^ 3)
. . .

```

Метод с именем `fnc` и формальными параметрами `a`, `x`, `z` используется при вычислении `c` и `d`.

**Пример 1.** Задан вектор `D(5)`. Создать класс `vect` осуществляющий ввод данных с клавиатуры, вывод элементов вектора на консоль, и перестановку первого по порядку элемента со `k`-тым отрицательным элементом вектора. Ограничение доступа в класс (снаружи можно вызывать только `public` части класса)

```

using System;
namespace ConsoleApplication1
{
class Program
{
    class vect          // класс векторов vect
    {
        double[] a; // объявление массива
        int n;      // размер массива
        public vect(int n)
        // конструктор создаёт вектор
        {
            this.a = new double[n];
            this.n = n;
        }
    }
}

```

```

public int vv()
// заполнение вектора целиком,
// клавиатурный ввод
{
    for (int i = 0; i < n; i++)
    {
        string buf;
        // строка символов
        buf = Console.ReadLine();
        // заполнение строки
        a[i] = Convert.ToDouble(buf);
        // преобразование строки в число
    }
    return (0);
}
public int vyv()
// вывод на консоль
// элементов вектора
{
    for (int i = 0; i < n; i++)
        Console.WriteLine("a["+i+"]=" +
            a[i]);
    return (0);
}
public int perest(int k)
// переставить 1 и

```

```

// k-ое отриц. число
{
// ключ; 0  если порядок
//      -1, если - нет
int k1 = 0;
// k больше размера вектора
if ( k >= n ) k1 = -1;
else
{
    // счетчик отрицательных чисел
    int cnt = 0;
    // указатель на позицию в массиве
    int pos = -1;
    double tmp;
    // подсчёт отрицательных чисел
    while ( cnt < k )
    {
        pos++;
        if (a[pos] < 0) cnt++;
    }
    tmp = a[pos];    // подсчёт
    a[pos] = a[0];   // перестановка
    a[0] = tmp;
    k1 = 0;
}
return k1;
}

```

```

// главная программа
static void Main()
{
    // создали
    vect d = new vect(5);
    d.vv();           // заполнили
    d.vyv();          // вывели на
                      // консоль
    d.perest(2);      // переставили
    d.vyv();          // вывели на
                      // консоль
}
}
}

```

**Пример 2.** Задана матрица  $D(3,3)$ . Написать метод транспонирования матрицы и метод перемножения квадратных матриц. Получить результат умножения транспонированной матрицы на соответствующую ей исходную. Отдельный класс для матриц не создаём. Работаем в классе `myprog`. Поэтому все содержимое класса доступно.

```

using System;
// моё пространство имён
namespace ConsoleApplication3
{
    class myprog

```

```
{
```

```
static void Main()
{
    // объявление матриц
    double[,] d = new double[3,3];
    double[,] dt = new double[3, 3];
    double[,] ddt = new double[3, 3];
    // заполнение d с клавиатуры
    int i; int j; // строки, колонки
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
            d[i, j] =
                Convert.ToDouble(Console.ReadLine());
    }
    // транспонирование матрицы
    transp(d, 3, dt);
    // умножение транспонированной
    // матрицы dt на исходную d
    mt_q_mult(dt, d, ddt, 3);
}

// транспонирование
static void transp(double[,] mtr,
int p, double[,] mtrt)
// double[,] mtr - так определяют матрицу в
// формальных параметрах
```

```

{
    int i; int j;
    for (i = 0; i < p; i++)
    {
        for (j = 0; j < p; j++)
            mtrt[j, i] = mtr[i, j];
    }
}

// умножение квадратных матриц
static void mt_q_mult(double[,] m_a,
double[,] m_b, double[,] m_r, int p)
// m_r=m_a * m_b , int p размер матриц
{
    int i; int j; int k; double mel;
    for (i = 0; i < p; i++) // строки m_a
    {
        // столбцы m_b
        for (j = 0; j < p; j++)
        {
            mel=0;
            // столбцы m_a и строки m_b
            for (k = 0; k < p; k++)
                mel=mel+m_a[i,k]*m_b[k,j];
            m_r[i,j]=mel;
        }
    }
}

```

```
}  
}
```

## 2. Практическая часть

### Задание к лабораторной работе

Написать программу, осуществляющую заданные вычисления с использованием процедур. Вид используемых классов и методов определить самостоятельно.

1. Заданы две матрицы  $A(4, 4)$  и  $B(4, 4)$ . Написать программу суммирования двух векторов  $X$  и  $Y$ , где  $X$  – вектор, в котором размещены элементы столбца матрицы  $A$  с минимальным средним арифметическим значением,  $Y$  – то же для матрицы  $B$ .
2. Заданы две матрицы  $A(4, 4)$  и  $B(4, 4)$ . Написать программу вычисления вектора  $Z = X + Y$ , где  $X$  – строка матрицы  $A$ , включающая минимальный элемент ее главной диагонали,  $Y$  – то же для матрицы  $B$ .
3. Заданы две матрицы  $A(4, 4)$  и  $B(3, 3)$  и два вектора  $C(4)$  и  $D(3)$ . Написать программу вычисления произведений матриц на соответствующие им вектора. Определить минимальное среднее арифметическое значение для полученных векторов.
4. Заданы две матрицы  $B(4, 4)$  и  $D(3, 3)$ . Написать программу транспонирования каждой из заданных матриц с последующим перемножением транспонированной матрицы на соответствующую ей исходную.

5. Заданы две матрицы  $A(3, 3)$  и  $B(3, 3)$ . Написать программу проверки – является ли произведение этих матриц перестановочным, т.е. проверить равенство  $AB = BA$ .
6. Заданы две матрицы  $C(4, 4)$  и  $D(3, 3)$ . Написать программу определения количества симметричных матриц. Матрица называется симметричной, если транспонированная матрица равна исходной. Для каждой симметричной матрицы вычислить сумму элементов, лежащих вне главной диагонали.
7. Заданы два массива  $C(6)$  и  $D(10)$ . Для каждого из них осуществить перестановку первого по порядку элемента со вторым отрицательным элементом массива.
8. Заданы три квадратных уравнения  $AX^2 + BX + C = 0$ ;  $DX^2 + EX + F = 0$  и  $ZX^2 + YX + S = 0$ . Написать программу нахождения максимального значения корня среди действительных корней этих уравнений.
9. Заданы два вектора  $A(0.052; 0.9; 0.15; 0.84; 0.67)$  и  $B(0.948; 0.1; 0.33; 0.16; 0.85)$ . Написать программу, позволяющую расположить элементы одного вектора по возрастанию, а другого – по убыванию. Вычислить сумму полученных векторов.
10. Заданы два двумерных массива  $A(4, 4)$  и  $B(3, 3)$ . Для каждого из них переставить столбцы с максимальным и минимальными элементами.
11. Заданы координаты четырёх точек  $A, B, C, D$  на плоскости. Определить наибольший из периметров треугольников  $ABC, ABD, ACD$ .

12. Три треугольника заданы координатами своих вершин. Написать программу вычисления площадей треугольников и определения минимальной площади.
13. Заданы два двумерных массива  $A(4, 4)$  и  $B(3, 3)$ . Для каждого из них переставить последнюю строку со строкой с максимальным средним арифметическим значением.
14. Заданы две матрицы  $A(4, 4)$  и  $B(4, 4)$ . Написать программу вычисления вектора  $Z = X - Y$ , где  $X$  – столбец матрицы  $A$ , включающий минимальный элемент матрицы;  $Y$  – то же для матрицы  $B$ .
15. Заданы три вектора  $X, Y, Z$ . Написать программу вычисления  $S = \sum_{i=1}^n C_i \cdot D_i$ , где  $C_i$  и  $D_i$  – компоненты векторов  $C$  и  $D$ , которые соответственно равны  $C = X + Y$ ;  $D = X - Z$
16. Заданы матрицы  $C(4, 4)$  и  $D(3, 3)$ . Определить индексы максимального элемента каждой из матриц среди элементов, расположенных выше главной диагонали.
17. Заданы вектора  $A(5), B(10), D(15)$ . Для каждого из них определить максимальный и минимальный элементы и их индексы.
18. Заданы координаты вершин трех треугольников. Определить сколько треугольников лежит внутри окружности радиуса  $R$  с центром в начале координат.
19. Заданы координаты десяти точек плоскости и координаты точки-полюса. Найти точку, максимально удалённую от полюса, среди первых четырёх заданных точек и такую же точку – среди последних шести. Определить также расстояние между найденными точками.

20. Заданы массивы  $Y(4, 4)$  и  $Z(8, 8)$ . Для каждого из них вычислить среднее арифметическое значение положительных элементов, расположенных выше главной диагонали.
21. Заданы координаты вершин трех треугольников. Определить треугольник с максимальным периметром.
22. Заданы три матрицы размерами  $2 \times 2$ ,  $3 \times 3$ ,  $4 \times 4$ . Для каждой из матриц определить среднее арифметическое значение положительных элементов главной диагонали.
23. Заданы три одномерных массива разной размерности. Для каждого из массивов определить повторяющиеся элементы.
24. Заданы координаты четырех точек на плоскости. Определить номера точек, расстояние между которыми минимальное.
25. Заданы две окружности  $(x - a)^2 + (y - b)^2 = r^2$  и  $(x - c)^2 + (y - d)^2 = R^2$  и координаты четырех точек. Определить количество точек, лежащих внутри каждой окружности.
26. Заданы три одномерных массива  $A(5)$ ,  $B(10)$ ,  $C(15)$ . Написать программу решения уравнения  $px^2 + dx + r = 0$ , где  $p$  – минимальный элемент матрицы  $A$ ,  $d$  – минимальный элемент матрицы  $B$ ,  $r$  – минимальный элемент матрицы  $C$ .
27. Заданы одномерные массивы  $X(5)$  и  $Y(7)$ . Для каждого из них определить количество и сумму элементов, которые без остатка делятся на заданное число  $B$ .
28. Составить программу заполнения массивов  $A(5)$  и  $B(10)$  факториалами значений индексов их элементов. Вычисление факториала выполнить в подпрограмме.

29. Заданы матрицы  $A(4,4)$ ,  $B(6,6)$ . Для каждой матрицы определить среднее арифметическое значение положительных элементов, расположенных не выше главной диагонали.
30. Матрицами  $F(2,2)$  и  $E(2,2)$  в декартовой системе координат заданы две окружности положением своего центра и точки на дуге. Вычислить параметры окружностей.